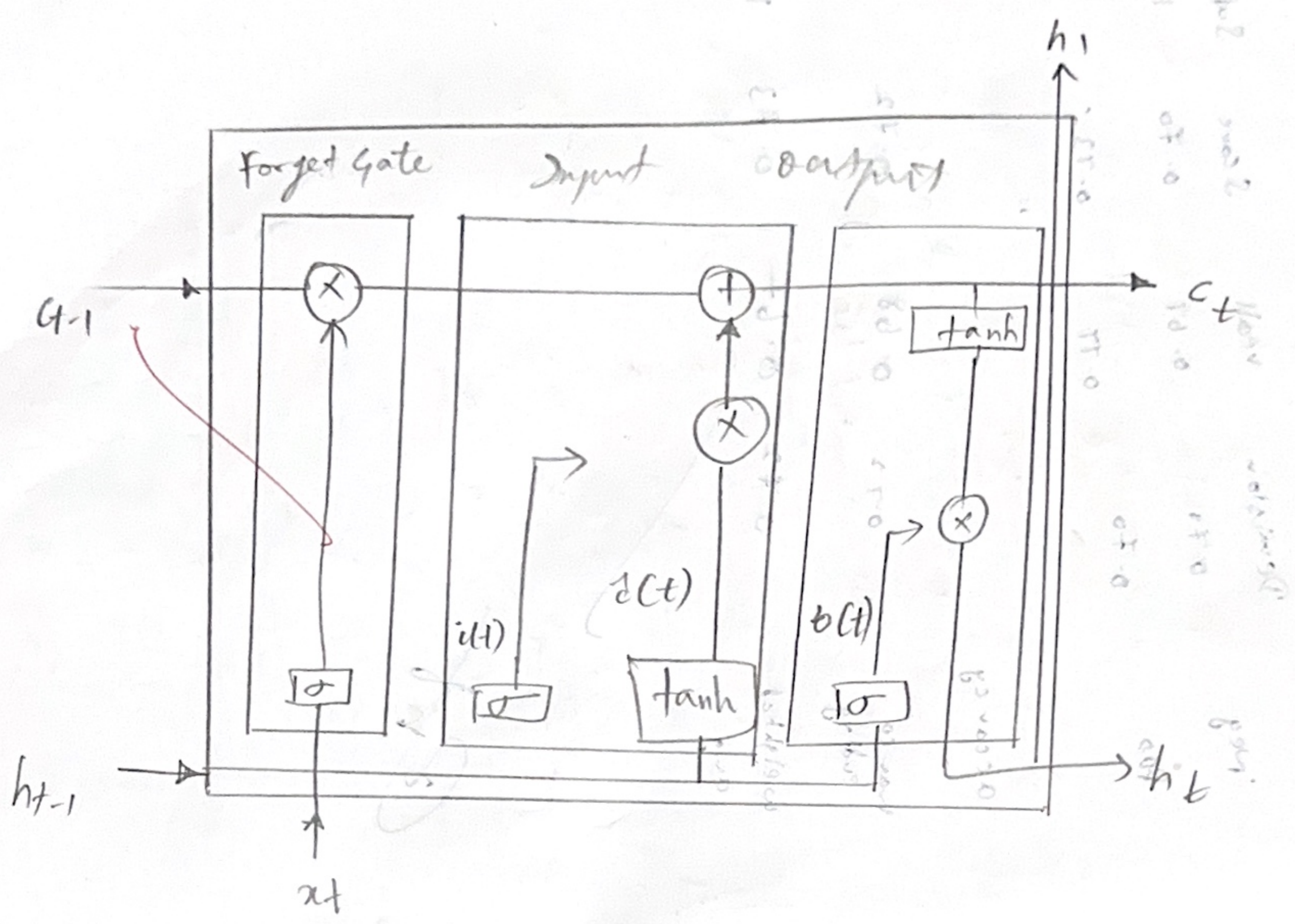


LSTM Architecture

[2019 2020]



9/10/25

Lab-8 Long Short Term Memory

Aim :-

To implement and train long short term memory network using Pytorch. by classifying the number sequences as either increasing decreasing.

Objective :-

To understand how LSTM Process sequential data.

To build and train an LSTM model for classification tasks.

To evaluate model Accuracy and Niballize Predictions.

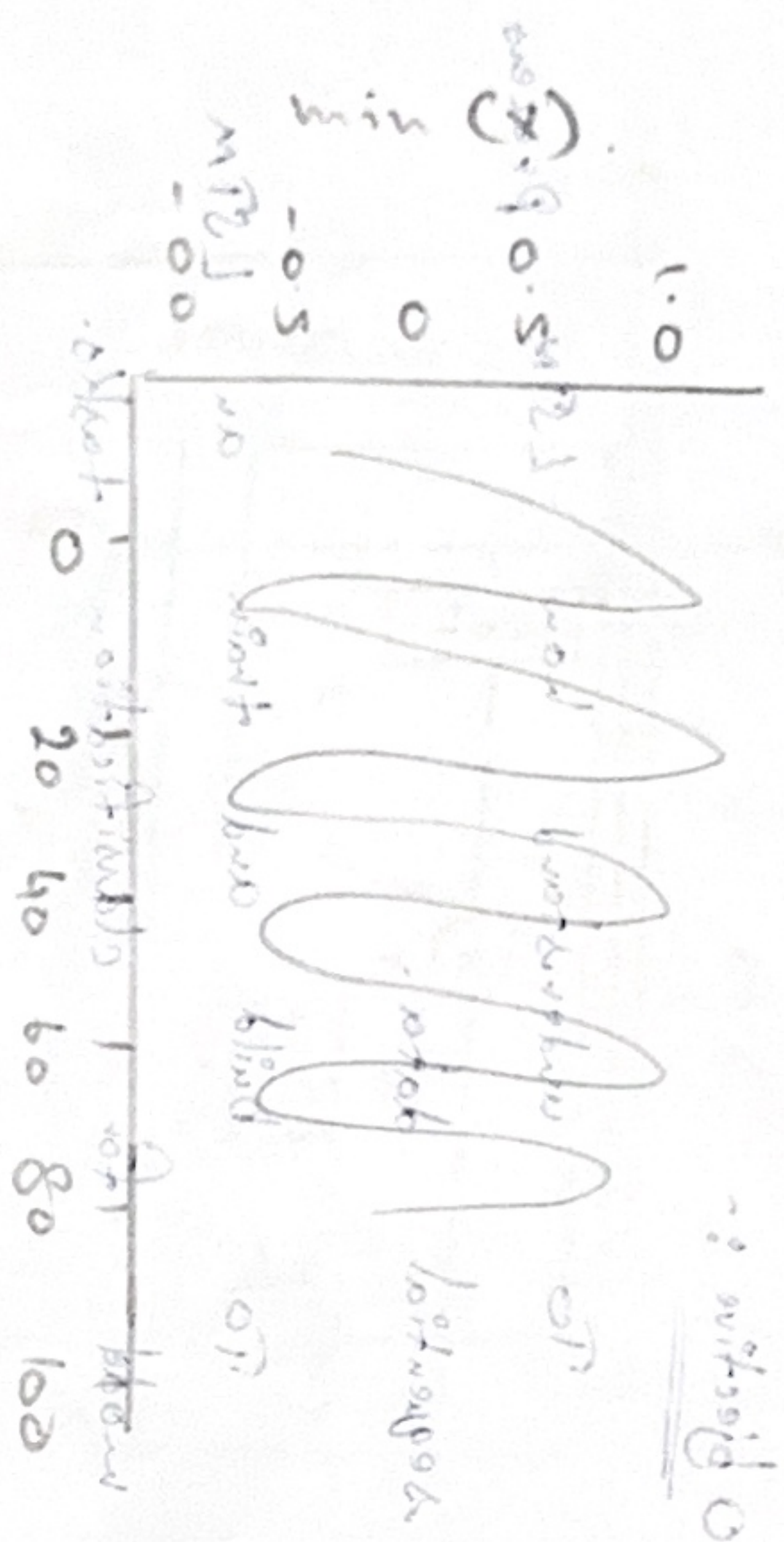
To analyze the Performance of LSTM

LSTM - LSTM Pattern and Recognition tasks.

2020-01-01 [0.0] 0.0
2020-01-01 [0.0] 0.0
2020-01-01 [0.0] 0.0
2020-01-01 [0.0] 0.0
2020-01-01 [0.0] 0.0

Project not too big 3-40

with tools good work but messy of it
 - GitHub id - Jupyter Cells Notebook browser
 after an assessment and run of it



Input shape : torch.Size [1960, 30, 1]
 Target shape : torch.Size [1960, 10, 1]

Training Loss : 0.03100

Epoch [1/30], loss : 0.079800
 Epoch [2/30], loss : 0.0462
 Epoch [3/30], loss : 0.0355
 Epoch [4/30], loss : 0.0291
 Epoch [5/30], loss : 0.0247
 Epoch [6/30], loss : 0.0212

Pseudocode:-

- Import necessary libraries.
- Generate Synthetic dataset.
- Pre process data.
- Define LSTM model. (Input-size, hidden size).
- Define loss function and optimizer.
- Cross entropy Loss.
- Adam optimizer.

→ Train the model.

- (i) For Pass.
- (ii) Compute loss.
- (iii) Backward Pass.
- (iv) Print loss and accuracy every few epochs.

Observation:-

Epoch	Training Loss	Training Accuracy (%)
10	0.6201	72.3
20	0.4052	80.1
40	0.2103	90.7
60	0.0982	95.5
80	0.0524	97.8
100	0.0310	99.0

